

Aplicação Prática de Aprendizado por Reforço e Imitation Learning com Unity ML-Agents

Lucas Brandão Rodrigues¹, Maria Carolina X. de Almeida¹,
Anna Pietra V. L. B. Moreira¹

¹Instituto de Informática – Universidade Federal de Goiás (UFG)
Goiânia – GO – Brasil

lucasbrandao4770@gmail.com

Abstract. Introduction: Reinforcement Learning (RL) and Imitation Learning (IL) are well-established Artificial Intelligence techniques with growing applications in digital games. However, the transition from their theoretical concepts to functional practical implementations remains a significant challenge for students, researchers, and developers, who lack systematic pedagogical support in the context of game development. **Objective:** This work proposes a structured pedagogical approach to facilitate this transition, enabling participants to build, train, and evaluate autonomous agents with the Unity ML-Agents toolkit, with emphasis on sharing solutions to recurring practical challenges. **Methodology or Steps:** The tutorial is organized into five modules with challenges of increasing complexity, conducted through an iterative problem → tool → solution cycle, spanning approximately four hours of hands-on activities — from RL fundamentals to the implementation of a complete project in a game environment, including IL techniques such as Behavioral Cloning (BC) and Generative Adversarial Imitation Learning (GAIL). **Expected Results:** Participants are expected to complete the tutorial able to independently implement autonomous agents with Unity ML-Agents, having experienced in practice the main configuration and training challenges. This hands-on experience grounds the academic contribution of this work: the systematic record of the lessons and practical bottlenecks documented throughout the tutorial aims to build a citable body of knowledge on the practical application of RL and IL in digital game development.

Keywords Reinforcement Learning, Imitation Learning, Unity ML-Agents, Game Development, Active Learning.

Resumo. Introdução: O Aprendizado por Reforço (Reinforcement Learning — RL) e o Aprendizado por Imitação (Imitation Learning — IL) são técnicas consolidadas da Inteligência Artificial com aplicações crescentes em jogos digitais. Contudo, a transição entre os conceitos teóricos dessas abordagens e sua implementação prática funcional permanece um desafio significativo para estudantes, pesquisadores e desenvolvedores, que carecem de suporte pedagógico sistemático no contexto do desenvolvimento de jogos. **Objetivo:** Este trabalho propõe uma abordagem pedagógica estruturada para facilitar essa transição, capacitando os participantes a construir, treinar e avaliar agentes autônomos com o toolkit Unity ML-Agents, com ênfase no compartilhamento de soluções para desafios práticos recorrentes.

Metodologia ou Etapas: O tutorial é organizado em cinco módulos com desafios de complexidade crescente, conduzidos por um ciclo iterativo de problema → ferramenta → solução, em aproximadamente quatro horas de atividades hands-on — dos fundamentos do RL à implementação de um projeto completo em um ambiente de jogo, passando por técnicas de IL como Clonagem Comportamental (Behavioral Cloning — BC) e Aprendizagem por Imitação Adversarial Generativa (Generative Adversarial Imitation Learning — GAIL). **Resultados Esperados:** Espera-se que os participantes concluam o tutorial aptos a implementar agentes autônomos com Unity ML-Agents de forma independente, tendo vivenciado na prática os principais desafios de configuração e treinamento. Essa vivência fundamenta a contribuição acadêmica do trabalho: o registro sistematizado das lições e “gargalos” documentados visa construir um corpo de conhecimento citável sobre a aplicação prática de RL e IL no desenvolvimento de jogos digitais.

Palavras-Chave Aprendizado por Reforço, Aprendizado por Imitação, Unity ML-Agents, Desenvolvimento de Jogos, Aprendizagem Ativa.

1. Introdução e Trabalhos Relacionados

O Aprendizado por Reforço (*Reinforcement Learning* - RL) constitui um paradigma fundamental da Inteligência Artificial, no qual agentes autônomos aprendem a tomar decisões através da interação com um ambiente, buscando maximizar recompensas acumuladas ao longo do tempo [Sutton e Barto 2018]. Este campo tem se consolidado como uma área de pesquisa cientificamente relevante, com aplicações que vão desde robótica até jogos digitais [Silver et al. 2017]. Os principais componentes estudados nesse contexto incluem a definição de espaços de estados e ações, o design de funções de recompensa e a escolha de algoritmos de otimização. Sobre essa base técnica comum, duas abordagens de aprendizado se destacam como complementares: o próprio RL, no qual o agente aprende por tentativa e erro a partir de sinais de recompensa, e o Aprendizado por Imitação (*Imitation Learning* - IL), no qual o agente acelera seu aprendizado a partir da observação de comportamentos demonstrados por um especialista [Hussein et al. 2017].

A aplicação prática dessas técnicas em ambientes de desenvolvimento de jogos requer a utilização de ferramentas e *frameworks* especializados que facilitem a prototipagem e a experimentação. Contudo, mesmo com a disponibilidade dessas ferramentas, a transição pedagógica entre os conceitos teóricos de RL e IL e a sua implementação prática funcional permanece um desafio significativo para estudantes, pesquisadores e desenvolvedores. Esse desafio se manifesta especialmente na configuração adequada de ambientes de treinamento, na definição de espaços de observação e ação apropriados, no balanceamento de funções de recompensa e na interpretação de métricas de desempenho — dificuldades que, embora recorrentes, carecem de um tratamento sistemático e orientado à prática no contexto do desenvolvimento de jogos.

Diversos trabalhos têm abordado aspectos dessa problemática. No âmbito mais amplo, [ElDahshan et al. 2022] conduziram uma revisão sobre o uso de RL em jogos digitais, identificando os principais desafios técnicos enfrentados por desenvolvedores e evidenciando que tais dificuldades persistem mesmo com o avanço das ferramentas

disponíveis. Nesse sentido, mesmo a disponibilidade de ambientes de treinamento bem estruturados e acessíveis, como o proposto por [Kurach et al. 2019], não elimina a complexidade inerente à configuração e validação de agentes autônomos — indício de que o obstáculo é de natureza pedagógica, e não apenas ferramental. No contexto específico do desenvolvimento de jogos, [Juliani et al. 2020] apresentaram o *toolkit* Unity ML-Agents como uma plataforma projetada para aproximar a pesquisa em RL de aplicações práticas, reduzindo a barreira de entrada para prototipagem e experimentação. Contudo, mesmo com o suporte dessa plataforma, [Ghasemipour et al. 2019] demonstraram que a escolha e aplicação de métodos de IL — como *Behavioral Cloning* (BC) e *Generative Adversarial Imitation Learning* (GAIL) [Ho e Ermon 2016] — exige compreensão aprofundada das diferenças entre abordagens, cujos impactos práticos no comportamento dos agentes não são trivialmente antecipados. Em conjunto, esses trabalhos evidenciam que, embora existam *frameworks* e metodologias estabelecidas, persiste uma lacuna entre o conhecimento teórico e a capacidade de implementação prática, particularmente no que se refere ao suporte pedagógico estruturado para superar os “gargalos” comuns encontrados durante o desenvolvimento de agentes autônomos em jogos digitais.

Neste trabalho, propomos uma abordagem pedagógica estruturada para facilitar essa transição utilizando o *toolkit* Unity ML-Agents, voltada a estudantes, pesquisadores e desenvolvedores com conhecimento em programação e familiaridade básica com Unity. A contribuição central é a sistematização de um processo *hands-on* de complexidade crescente, documentando lições e soluções práticas que vão além da replicação da documentação oficial.

2. Conteúdo e Didática

Em resposta à lacuna pedagógica identificada entre o conhecimento teórico e a implementação prática de RL e IL no desenvolvimento de jogos, o tutorial é estruturado em cinco módulos de complexidade crescente, totalizando aproximadamente quatro horas de duração, conforme sintetizado na Tabela 1. A dinâmica pedagógica adotada é fundamentalmente *hands-on* e baseada em projetos, priorizando a aprendizagem ativa [Prince 2004, Freeman et al. 2014] por meio de um ciclo iterativo de *problema* → *ferramenta* → *solução*. Nessa abordagem, cada módulo é introduzido por um desafio concreto, seguido da apresentação dos conceitos e ferramentas necessários para resolvê-lo, culminando na implementação prática pelos participantes. Ao final de cada módulo principal, são previstos Momentos de Discussão estruturados, nos quais os participantes são encorajados a expor dúvidas, compartilhar dificuldades e trocar experiências — etapa considerada essencial para a consolidação do conhecimento e para o levantamento dos “gargalos” mais recorrentes na prática.

O Módulo 0, de caráter introdutório, nivela o ponto de partida do grupo e é acessível mesmo a participantes sem experiência prévia com o *toolkit* ML-Agents ou com os conceitos de RL. Além da introdução teórica — agente, ambiente, estado, ação e recompensa —, inclui uma etapa de verificação e configuração do ambiente de desenvolvimento, garantindo que todos os participantes iniciem as atividades práticas em condições equivalentes.

O Módulo 1 representa o primeiro ciclo completo da dinâmica proposta. O “Hello World” do ML-Agents, um agente que aprende a alcançar um alvo, foi escolhido por

Tabela 1. Estrutura dos módulos do tutorial.

Módulo	Duração	Desafio proposto	Técnicas e ferramentas
0 — Introdução	30 min	Nivelamento e configuração do ambiente	Conceitos de RL; Unity; ML-Agents
1 — Primeiro agente	60 min	Agente que aprende a alcançar um alvo (“Hello World”)	Espaços de observação e ação; função de recompensa; TensorBoard
2 — Imitação	60 min	Tarefa elaborada a partir de demonstrações humanas	Gravação de demonstrações; BC; GAIL
3 — Projeto final	75 min	Agente autônomo em jogo ou ambiente simulado complexo	Integração RL+IL; <i>design</i> de recompensas; análise de comportamento
4 — Encerramento	15 min	Síntese e próximos passos	Repositório GitHub; guia de referência

expor de imediato as decisões de *design* mais impactantes: a definição dos espaços de observação e ação, a implementação da função de recompensa e a interpretação dos gráficos de treinamento via TensorBoard.

O Módulo 2 amplia a complexidade ao introduzir o IL. A experiência de controlar manualmente o agente para gerar demonstrações de comportamento é deliberada: ela torna concreto o custo de produção dos dados de imitação antes de apresentar as técnicas de BC e GAIL. A análise comparativa dos resultados de ambas as abordagens incentiva a reflexão crítica sobre as escolhas de *design* — dimensão pedagógica que vai além da simples execução técnica.

O Módulo 3 constitui o projeto final do tutorial. Ao exigir que os participantes projetem, implementem e treinem um agente autônomo em um ambiente complexo, este módulo transfere para eles a responsabilidade pelas decisões de *design* — sobre observações, ações e recompensas — e pela interpretação do comportamento emergente. A maior duração reflete intencionalmente o espaço reservado para autonomia e experimentação.

Por fim, o Módulo 4 encerra o tutorial com uma síntese das técnicas desenvolvidas, sugestões de projetos futuros e algoritmos mais avançados, e uma sessão de perguntas e respostas. Os participantes recebem acesso a um repositório no GitHub contendo os projetos Unity finalizados, códigos-fonte comentados e um guia de referência — material projetado para sustentar a continuidade do aprendizado para além do evento.

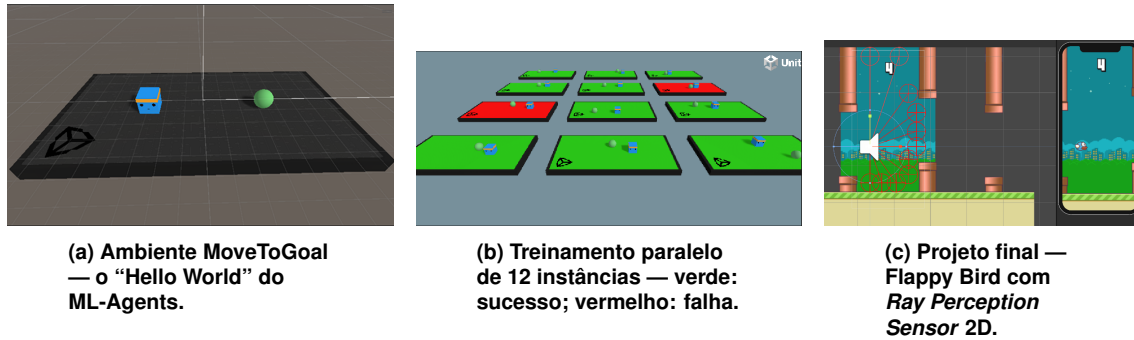


Figura 1. Progressão dos ambientes práticos do tutorial.

3. Discussão e Contribuições Esperadas

O presente trabalho oferece contribuições em duas frentes complementares: acadêmica e pedagógica. Do ponto de vista acadêmico, o tutorial sistematiza e documenta um conjunto de práticas, observações e soluções para desafios recorrentes no desenvolvimento de agentes autônomos com Unity ML-Agents — conhecimento que, embora relevante, raramente aparece de forma estruturada na literatura científica, e que aqui é registrado a partir da experiência prática direta dos autores com os desafios de implementação descritos. Ao registrar essas lições, o trabalho contribui para a construção de um corpo de conhecimento citável e permanente sobre os “gargalos” práticos do uso de RL e IL no desenvolvimento de jogos, complementando trabalhos anteriores que, em sua maioria, abordam os aspectos algorítmicos dessas técnicas sem aprofundar os desafios de implementação enfrentados por desenvolvedores e pesquisadores em formação. Esse registro está consolidado em um repositório público no GitHub¹ — contendo os projetos completos, os códigos-fonte comentados e um guia de referência — que constitui o artefato permanente e citável associado ao tutorial.

Do ponto de vista pedagógico, a principal contribuição reside na estruturação de uma metodologia de ensino ativa e orientada a projetos, organizada em torno do ciclo *problema* → *ferramenta* → *solução*. Essa abordagem se diferencia de tutoriais convencionais por não se limitar à replicação da documentação oficial do *toolkit*, mas por conduzir os participantes por uma progressão deliberada de desafios de complexidade crescente, na qual cada módulo constrói sobre o anterior. Os Momentos de Discussão estruturados ao final de cada módulo reforçam essa dimensão pedagógica, criando espaços intencionais para a reflexão coletiva e para a troca de experiências entre participantes com diferentes perfis e trajetórias. Dessa forma, o tutorial não apenas capacita tecnicamente os participantes, mas também os instrumentaliza para que continuem desenvolvendo autonomia na área.

A relevância dessa contribuição é ampliada pelo perfil diverso do público esperado no SBGames, que reúne estudantes de graduação e pós-graduação, pesquisadores e desenvolvedores da indústria de jogos. Ao oferecer um ponto de entrada estruturado e

¹<https://github.com/lucasbrandao4770/sbgames2026-tutorial-mlagents>

acessível para o uso prático de RL e IL com Unity ML-Agents, o tutorial contribui para a democratização do acesso a essas técnicas no contexto latino-americano, alinhando-se ao papel do SBGames como principal fórum de integração entre academia e indústria de jogos na região.

Apesar das contribuições, o formato presencial de quatro horas, a dependência de infraestrutura de laboratório e a heterogeneidade de experiências prévias dos participantes constituem limitações que condicionam a replicabilidade e o ritmo individual de aprendizado.

Referências

- ElDahshan, K. A., Farouk, H., e Mofreh, E. (2022). Deep reinforcement learning based video games: A review. In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 302–309.
- Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., e Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23):8410–8415.
- Ghasemipour, S. K. S., Zemel, R. S., e Gu, S. S. (2019). A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*.
- Ho, J. e Ermon, S. (2016). Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, volume 29, pages 4565–4573.
- Hussein, A., Gaber, M. M., Elyan, E., e Jayne, C. (2017). Imitation learning: A survey of learning methods. *ACM Comput. Surv.*, 50(2).
- Juliani, A., Berges, V.-P., Teng, E., Cohen, A., Harper, J., Elion, C., Goy, C., Gao, Y., Henry, H., Mattar, M., e Lange, D. (2020). Unity: A general platform for intelligent agents.
- Kurach, K., Raichuk, A., Stańczyk, P., Zajac, M., Bachem, O., Espeholt, L., Riquelme, C., Vincent, D., Michalski, M., Bousquet, O., e Gelly, S. (2019). Google research football: A novel reinforcement learning environment.
- Prince, M. (2004). Does active learning work? a review of the research. *Journal of Engineering Education*, 93(3):223–231.
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., e Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359.
- Sutton, R. S. e Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 2nd edition.